

⑫ 公開特許公報 (A)

昭63-115004

⑬ Int. Cl. \*

G 01 C 21/00  
G 09 B 29/10

識別記号

庁内整理番号

N-6666-2F  
A-8302-2C

⑭ 公開 昭和63年(1988)5月19日

審査請求 未請求 発明の数 1 (全15頁)

⑮ 発明の名称 車両の現在地推測方法

⑯ 特 願 昭61-260420

⑰ 出 願 昭61(1986)10月31日

⑱ 発 明 者 柏 崎 隆 埼玉県所沢市花園4丁目2610番地 バイオニア株式会社所  
沢工場内  
⑲ 発 明 者 安 藤 齊 埼玉県所沢市花園4丁目2610番地 バイオニア株式会社所  
沢工場内  
⑳ 発 明 者 細 井 雅 幸 埼玉県所沢市花園4丁目2610番地 バイオニア株式会社所  
沢工場内  
㉑ 発 明 者 福 島 敦 彦 埼玉県所沢市花園4丁目2610番地 バイオニア株式会社所  
沢工場内  
㉒ 出 願 人 バイオニア株式会社 東京都目黒区目黒1丁目4番1号  
㉓ 代 理 人 弁理士 藤村 元彦

明 細 書

1. 発明の名称

車両の現在地推測方法

2. 特許請求の範囲

地図の位置情報を数値データとして記憶しておき、車両の進行方位を検出する方位センサ及び車両の走行距離を検出する走行距離センサの各出力データに基づいて車両の現在地を推測して地図上に表示し、進行方位の道路データ部分に対するずれ角及び推測位置から最近傍線分までの距離がそれぞれ所定範囲内であれば、前記道路データ部分上に現在地を修正する車載ナビゲーション装置における車両の現在地推測方法であって、前記ずれ角又は前記距離が前記所定範囲外になったことを検出し、その地点からの走行距離を算出し、この走行距離に比例した距離に関し、前記最近傍線分までの距離のサンプリング値を用いて走行データと道路データの相関をとり、相関性が高ければ前記道路データ上に現在地を修正することを特徴と

する車両の現在地推測方法。

3. 発明の詳細な説明

技術分野

本発明は、車載ナビゲーション装置における車両の現在地推測方法に関するものである。

背景技術

近年、地図情報をメモリに記憶しておき、その地図情報をメモリから読み出して車両の現在地とともに表示装置に表示させることにより、車両を所定の目的地に誘導する車載ナビゲーション装置が研究、開発されている。

かかるナビゲーション装置では、車両に搭載された走行距離センサや方位センサ等の出力データに基づいて車両の走行距離や方位等を検出し、これに基づいて随々刻々と変化する車両の現在地を推測することにより、ディスプレイに画送されている地図上への現在地の表示が行なわれる。

ところで、道路地図を数値化するに際しては、車両が走行可能な道路のすべてに関して数値化することが望ましいが、センサ類の精度等により、

すべての道路に同じ正確にナビゲーションを行なうことは困難である。また、数値化する作業も膨大なものとなる。したがって、あるレベルまで数値化を限定せざるを得なく、この場合数値化されていない道路を走行することを想定しなければならない。

#### 発明の概要

本発明は、上述した点に鑑みなされたもので、道路地図の数値化のレベルに拘らず、誤りの無いナビゲーションの実現を可能とした車両の現在地推測方法を提供することを目的とする。

本発明による車両の現在地推測方法は、進行方位の道路データ段分に対するずれ角又は推測位置から最近傍段分までの距離が所定範囲外になったことを検出し、その地点からの走行距離を算出し、この走行距離に比例した距離に関し、最近傍段分までの距離のサンプリング値を用いて走行データと道路データの相関をとり、相関性が高ければ前記道路データ上に現在地を修正することの特徴としている。

U7の各種の処理プログラムやその他必要な情報が予め書き込まれたROM(リード・オンリ・メモリ)8と、プログラムを実行する上で必要な情報の書き込み及び読み出しが行なわれるRAM(ランダム・アクセス・メモリ)9と、いわゆるCD-ROM、ICカード等からなり、デジタル化(数値化)された地図情報が記録された記録媒体10と、V-RAM(Video RAM)等からなるグラフィックメモリ11と、CPU7から送られてくる地図等のグラフィックデータをグラフィックメモリ11に描画しかつ画像としてCRT等のディスプレイ12に表示すべく制御するグラフィックコントローラ13とから構成されている。入力装置14はキーボード等からなり、使用者によるキー入力により各種の指令等をシステムコントローラ5に対して発する。

記録媒体10には地図情報が記録されるのであるが、そのデータ構造について以下に説明する。先ず、第2図(A)に示すように、日本全国を例えば16384(=2<sup>14</sup>)[m]四方のメッシュ

以下、本発明の実施例を図に基づいて詳細に説明する。

第1図は、本発明に係る車載ナビゲーション装置の構成を示すブロック図である。同図において、1は地磁気に基づいて車両の方位データを出力するための地磁気センサ、2は車両の角速度を検出するための角速度センサ、3は車両の移動距離を検出するための走行距離センサ、4は経度及び経度情報等から車両の現在地を検出するためのGPS(Global Positioning System)装置であり、これら各センサ(装置)の出力はシステムコントローラ5に供給される。

システムコントローラ5は、各センサ(装置)1~4の出力を入力としA/D(アナログ/デジタル)変換を行なうインターフェース6と、種々の画像データ処理を行なうとともにインターフェース6から順次送られてくる各センサ(装置)1~4の出力データに基づいて車両の移動量等を演算するCPU(中央処理回路)7と、このCP

に分割し、このときの1つのメッシュをテリトリと呼ぶ。テリトリはテリトリNo. (Tx, Ty)で識別され、各テリトリには例えば図の左下のテリトリを基準にテリトリNo. が付与される。テリトリNo. は現在地(Crntx, Crnty)より求まる。テリトリは本データ構造の中で最も大きな管理単位となる。地図データファイル全体の構成が第2図(B)に示されており、テリトリIDファイルには、第2図(C)に示すように、テリトリNo. (Tx, Ty)のファイルにおける先頭アドレス、テリトリの左下の経度(実数)、テリトリの左下の経度(実数)、地磁気の偏角(実数)等のデータが各テリトリ毎に書き込まれている。

テリトリファイルは本データ構造において最も重要なファイルであり、各種の地図データや地図描画に必要なデータが書き込まれている。第3図(A)において、ナビID及びセクションテーブルがナビゲーションにおける道路及び交差点検索用ファイル、ピクチャーIDが表示管理用ファ

イル、道路セクションデータから交差点データまでが実際の地図データである。地図データは、第3図(B)に示すように、地図構造となっており、最下層が川、湖、湖等のポリゴンデータ、その上が道路、鉄道等のラインデータ、その上が各種マーク等のキャラクタデータ、その上が地名等の文字データ、そして最上層が交差点データとなっている。最上層の交差点データは後述する交差点引込みのために用いられるデータであり、ディスプレイ上には表示されない。

次に、第4図(A)に示すように、1つのテリトリーの中を例えば256分割し、これにより得られる1024(2<sup>8</sup>)[m]四方のメッシュをユニットと呼ぶ。このユニットも同様にユニットNo. (Nx, Ny)で管理され、そのNo. (Nx, Ny)は現在地(Crntx, Crnty)より求まる。ユニットは中間的な管理単位で、地図情報はこの単位で記録され、ユニットが256個集まってテリトリーファイルを構成する。地図描写の際はこの単位を基に行なわれるので、描画の基

本単位とすることができる。ナビDファイルには、第4図(B)に示すように、ユニットNo. (Nx, Ny)のファイルにおけるライン先頭アドレス、交差点先頭アドレス、道路セクション先頭アドレス、交差点先頭アドレス等のデータが各ユニット毎に格納されている。

更に、第5図(A)に示すように、1つのユニット内を例えば16分割し、これにより得られる256(2<sup>8</sup>)[m]四方のメッシュをセクションと呼ぶ。このセクションも同様にセクションNo. (Sx, Sy)で管理され、そのNo. (Sx, Sy)は現在地(Crntx, Crnty)より求まる。セクションは最も小さい管理単位であり、この範囲内の線分(線分の端で道路等が表わされる)や交差点の情報が第5図(B), (C)に示す如くセクションテーブルとして、更に第6図(A), (B)及び第7図(A), (B)に示す如くセクションデータとしてテリトリーファイルに登録されている。

また、第3図(A)に示すように、テリトリー

ファイル内には表示管理用のピクチャーIDというファイルがある。本実施例では、地図データの縮尺が例えば2.5万分の1、5万分の1、10万分の1の3種類に設定されており、実際の地図データとしては、最も縮尺の大きい2.5万分の1のものだけを持っている。各縮尺の地図は、第8図～第10図の各図(A)に示すように、エリアに分割され、このエリアはエリアNo. (Anx, Any)で管理される。エリアNo. (Anx, Any)は現在地(Crntx, Crnty)より求まる。縮尺が2.5万分の1の場合、エリアNo. とユニットNo. は同じであり、5万分の1の場合は1つのエリアがユニットファイル4個分となり、10万分の1の場合は1つのエリアがユニット16個分となる。また、各縮尺のピクチャーIDには、第8図～第10図の各図(B)にそれぞれ示すように、その縮尺の地図を表示するのに必要なポリゴン、ライン、キャラクタ、文字データの先頭アドレスとデータサイズが記録されている。

続いて、ポリゴンデータとラインデータについ

て説明する。ポリゴンデータとラインデータは、第11図(A)及び第12図(A)に示すように、始点と終点で表わされる線分のあるベクトル(線分)で表わされる。ここで、最も縮尺の大きい2.5万分の1の地図データで5万分の1や10万分の1の地図を表現すると、始点・終点間が縮まるのでディスプレイ上で見た限りでは、全ての点を表示しなくても差し支えないことがある。このことを考慮に入れて、ディスプレイ上に表示した場合に、見た目上省略しても差し支えない点の情報を、第11図(B)及び第12図(B)に示すように、予めポリゴン及びラインデータの各間引きビットに入れておく。そして、各縮尺の表示時に間引きビットをチェックして必要に応じて間引きビットに情報が入っている点を除く、いわゆる間引きを行なうことにより、表示する線分(ベクトル)数を減らすことができる。

また、第13図(A)に示すように、1ユニット内に存在する交差点の全てに通し番号(x, y, n)が付されている。ところで、交差点には、西交

型、Y字路、5叉路等々あるが、特に方位の似た道路が複数入っている交差点では、この交差点を通過したときに、センサの精度、計算誤差、地図精度等により道路の選択を誤り、ディスプレイ上に現在地が表示されている道路と実際に走行している道路とが一致しない状態が生ずる可能性がある。そこで、このような交差点に対しては、第13図(B)に示すように、交差点の難易度を示す難易度データを交差点データ中の難易度ビットにいておく。そして、交差点を通過するときには、この難易度データに基づいた処理を行なえば、誤った道路の選択を防止できるのである。

次に、地図データの表示に関して、グラフィックメモリ11として例えばV-RAMを用いた場合について説明する。表示の構成としては、第14図(A)に示すように、512(ドット)×512(ドット)のV-RAM上で画面を16分割し、それぞれのエリアに独立した1枚の地図を表示するようにする。1エリアは128(ドット)×128(ドット)の1ユニットであり、更に1

6分割することにより1エリアは32(ドット)×32(ドット)の1セクションとなる(第14図(B)、(C)を参照)。実際の地図ディスプレイには、第14図(A)の中央の4画面に相当する256(ドット)×256(ドット)のエリア(太線で囲ったエリア)が表示され、このエリアがV-RAM上を移動することによって車両の現在地の動きを表現する。

次に、CPU7によって実行される基本的な手順を第15図のフローチャートに従って説明する。

CPU7は、先ず最初にプログラムを実行させるためのイニシャライズを行ない(ステップS1)、しかる後車両の現在地がセットされているか否かを判断する(ステップS2)。現在地がセットされていない場合は、現在地セットルーチンの実行(ステップS3)、例えば入力装置14でのキー入力による現在地のセットが行なわれる。次に、走行距離を零にし(ステップS4)、続いて入力装置14からのキー入力があるか否かの判断を行なう(ステップS5)。

キー入力が無い場合は、ディスプレイ12上に現在地周辺の地図表示を行なうとともに車両の現在位置及びその方位を例えば車両マークにてこの地図上に表示し、車両が移動したらその移動に伴い地図をスクロールさせ、更に現在グラフィックメモリ11上にある地図データの範囲を車両位置が越えそうときには、記録媒体10から必要な地図データを読み出してディスプレイ12上に表示する(ステップS6)。

キー入力があると、その入力データに応じて現在地の再セット(ステップS7)、センサ補正(ステップS8)、目的地セット(ステップS9)及び地図の拡大・縮小(ステップS10)の各ルーチンを実行する。

また、CPU7はタイマーによる割込みにより、第16図に示すように、一定時間間隔で地磁気センサ1及び角速度センサ2の各出力データに基づいて車両の方位を常に計算する処理を行なう(ステップS11、S12)。

CPU7は更に、走行距離センサ3よりデータ

が入力された場合は、走行距離センサによる割込み処理を行なう。この割込み処理では、第17図に示すように、走行距離と方位からの現在地の口出(ステップS13)、右折、左折の判定(ステップS14)、道路への引込み(ステップS15)、交差点引込み(ステップS16)、走行距離による引込み(ステップS17)が実行される。なお、このステップS13～ステップS17における各処理に関しては、後で詳細に説明する。

また、GPS装置4より与えられる精度、経度データは、第18図に示すように、GPSデータ受信割込みにより処理され、現在地データとして座標変換される(ステップS18)。

車両の走行距離は走行距離センサ3の出力から求められる。この走行距離センサ3としては、例えば、車のいわゆるスピードメータケーブルの回転数(JIS規格では、637回転/Km)より1回転の距離を積分することにより走行距離を求める構成のものが用いられるが、センサ3の精度により与えられる走行距離に誤差が生ずることは避

けられない。また、センサ3の精度だけではなく、地図の精度、タイヤの空気圧の変化、低スリップ等も走行距離の誤差の要因となる。従って、走行距離の補正を度々行なわないと、正確に距離を求めることができなくなってしまうことになる。このため、走行距離センサ3の出力より得た実際の距離と地図データより得た距離とから距離補正係数 $r_s$ を求め、この補正係数 $r_s$ を用いて距離補正を行なうことにより、走行距離を常に正確に検出できるのである。

また、車両の方位は地磁気センサ1の出力から求められる。この方位検出方法に関しては、本出願人等による特願昭60-282341号明細書等に記載されている。この地磁気センサ1の示す北は磁北であり、地図北ではない。このため、磁北が地図北に対してずれていた場合、第19図に示すように、基準位置から一定距離だけ走行したときに地磁気センサ1の出力から得られる推定現在地 $P_1$ は実際の現在地 $P_2$ に対してずれを生じることになる。そのため、地磁気センサ1より求

めた方位を地図方位に変換する作業が必要となる。この変換作業は、第20図に示すように、2次元幾何の座標変換で求まる回転角、即ち方位補正係数 $\theta_s$ によって行なわれる。この方位補正係数 $\theta_s$ は地域により変化し、更に地磁気センサ1を車体に取り付けたときに生じる取付け誤差によっても変わる。この方位補正係数 $\theta_s$ は、第21図に示すように、当該係数を常として位置のわかっている2地点を走行し、慣性航法により求められた現在地と到着点との誤差により求めることができる。この方位補正係数 $\theta_s$ を用いて方位補正を行なうことにより、車両の方位を常に正確に検出できるのである。

なお、距離補正係数 $r_s$ 及び方位補正係数 $\theta_s$ の検出方法は、本出願人等による特願昭60-282344号明細書等に記載されている。

車両の走行時には、最近傍交差点への引込みや最近傍線分への引込みが適宜行なわれるのであるが、この引込みを行なうためには、現在地に最も近い道路（最近傍線分）や交差点（最近傍交差点）

を探し出す作業が必要となる。この最近傍交差点や最近傍線分をサーチする作業は、線分や交差点データの量が多い、即ちサーチエリアが広いと、時間がかかってしまい、時々刻々と変化する現在地をスムーズに表示できないことになる。ところが、本実施例においては、第2図～第5図に示したデータ構造から明らかなように、現在地からのサーチエリアをできるだけ小さくし、かつそのエリアに入る線分や交差点のデータを管理するデータ（セクションデータ、セクションテーブル）を持たせていることにより、最小単位のセクションをサーチエリアとしてその中から線分や交差点をサーチすることができるので、サーチに要する時間を短縮できるのである。以下、CPU7によって実行される現在地から最近傍線分と最近傍交差点をサーチする手順を、第22図のフローチャートに従って説明する。

CPU7はまず、現在地（ $Cnrx$ ,  $Cnry$ ）からテリトリノ（ $Tx$ ,  $Ty$ ）、ユニット $N$ （ $Nx$ ,  $Ny$ ）、セクション $N$ （ $Sx$ ,

$Sy$ ）をそれぞれ求める（ステップS20～S22）。これは、各エリアが $2^n$ 単位で分割されているので、簡単な旗印（割印）で求めることができる。次に、セクションをサーチエリアとして、この中に存在する線分と交差点データをセクションテーブルとセクションデータを参照することによりロードする（ステップS23～S25）。ロードしたデータを基に、現在地からサーチエリア内の全ての線分までの距離（線分に対する垂線の長さ）、全ての交差点までの距離を計算し、それらと比較することによって最近傍線分と最近傍交差点を得ることができる（ステップS26）。サーチを行なう際のスピードは、線分の本数や交差点の個数に比例するが、前述したデータ構造に基づくサーチ方式によれば、サーチエリア（セクション）が小さく、計算の対象となる線分の本数や交差点の個数が少ないので、高速サーチが可能となるのである。

続いて、CPU7によって実行される線分引込みルーチン及び交差点引込みルーチンの手順を、

第23図及び第24図のフローチャートに従って説明する。

第23図の差分引込みルーチンは一定走行距離、例えば20[m]動に行なわれる。CPU7は先ず、第22図の手順に従って最近傍差分しをリーチし、その差分しについて、現在地から差分しまでの距離 $l$ 及び差分しの地図座標とす角度 $\theta_c$ を求める(ステップS30)。そして自車角度 $\theta$ と上記角度 $\theta_c$ との差の絶対値と所定のスレシヨルド値 $\theta_{lh}$ とを比較し(ステップS31)、スレシヨルド値 $\theta_{lh}$ よりも大きい場合は、引込みを行なうための有効差分が存在せず、しかも道路から外れたと見なす(ステップS32)。

スレシヨルド値 $\theta_{lh}$ 以下の場合は、次に上記距離 $l$ と所定のスレシヨルド値 $l_h$ とを比較し(ステップS33)、距離 $l$ がスレシヨルド値 $l_h$ よりも小さいか等しい場合は、その差分しは有効差分であり道路上であると見なし(ステップS34)、差分上に引込みを行なって現在地の修正を行なう(ステップS35)。また、距離 $l$

がスレシヨルド値 $l_h$ よりも大きい場合には、有効差分は存在するが道路から外れているものと判断する(ステップS36)。

一方、第24図の交差点引込みルーチンは交差点フラグがオンのときに行なわれる。CPU7は先ず、第22図の手順に従って最近傍交差点をリーチし、その交差点について交差点までの距離 $l_c$ (ステップS40)。そして距離 $l_c$ と所定のスレシヨルド値 $l_h$ とを比較し(ステップS41)、距離 $l_c$ がスレシヨルド値 $l_h$ よりも小さいか等しい場合は、引込みを行なうべき交差点が有ると判断し(ステップS42)、その交差点に現在地を修正する(ステップS43)。距離 $l_c$ がスレシヨルド値 $l_h$ よりも大きい場合は、引込みを行なうべき交差点が無いものとして引込みを行わない(ステップS44)。

次に、道路データから外れた所を走行し、再度道路データ近辺を走行した場合、走行データと道路データとの相関をとり、相関性が高ければ道路データに引き込む、いわゆるパターンマッチング

ルーチンについて、第25図及び第26図のフローチャートに従って説明する。

第25図は、サンプルされたデータを配列にセーブする距離差データセーブルーチンであり、タイマー割込みによって実行される。CPU7は先ず、道路引込み中であるか否かを判断し(ステップS50)、引込み中でなければ、道路から外れてからの走行距離を表わすカウンタのカウント値Nocをインクリメントする(ステップS51)。次に、最近傍差分なる有効差分が有るか否かを判断し(ステップS52)、有効差分が有る場合は、最近傍差分までの距離 $E(i)$ と前回求めた距離 $E(i-1)$ との差分 $D(i)$ を配列にセーブする(ステップS53)。有効差分が無い場合は、1回毎に極性を逆にした大きな数値をダミーデータとして配列にセーブする(ステップS54)。そしてポインタを"1"だけ進め(ステップS55)、しかる繰りターンする。

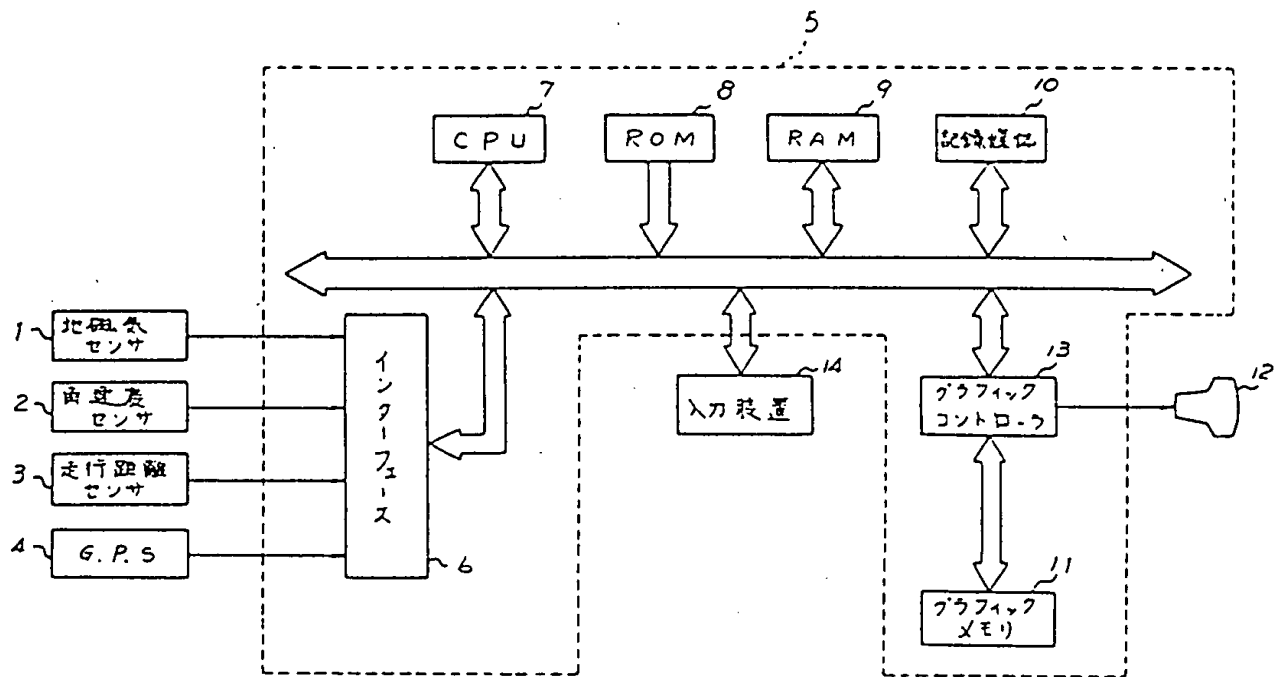
第26図は、パターンマッチングルーチンであり、メインルーチンから所定のタイミングで呼び

出されるものである。CPU7は先ず、引込み中であるか否かを判断し(ステップS60)、引込み中であれば、マッチングの必要がないので、そのままリターンする。次に、有効差分が有るか否かを判断し(ステップS61)、有効差分がなければ、道路データから一定距離以上離れて走っている、そのままリターンする。続いて、最小単位距離以上走行したか否かを判断し(ステップS62)、走行していない場合は、そのままリターンする。

次に、道路から外れてからの走行距離が一定の距離内であるか否かを判断し(ステップS63)、一定の距離内である場合は、地磁気センサ補正ルーチンを実行する(ステップS64)。この地磁気センサ補正ルーチンでは、車両が例えば路切を通過した際、車両ボデーが磁場を受けると、地磁気センサ1の出力がずれて道路データに乗らなくなる可能性がある、地磁気センサ1の出力のずれを検出し、それに基づいて地磁気センサ1の出力データの補正が行なわれるのである。



第 1 回



第2圖

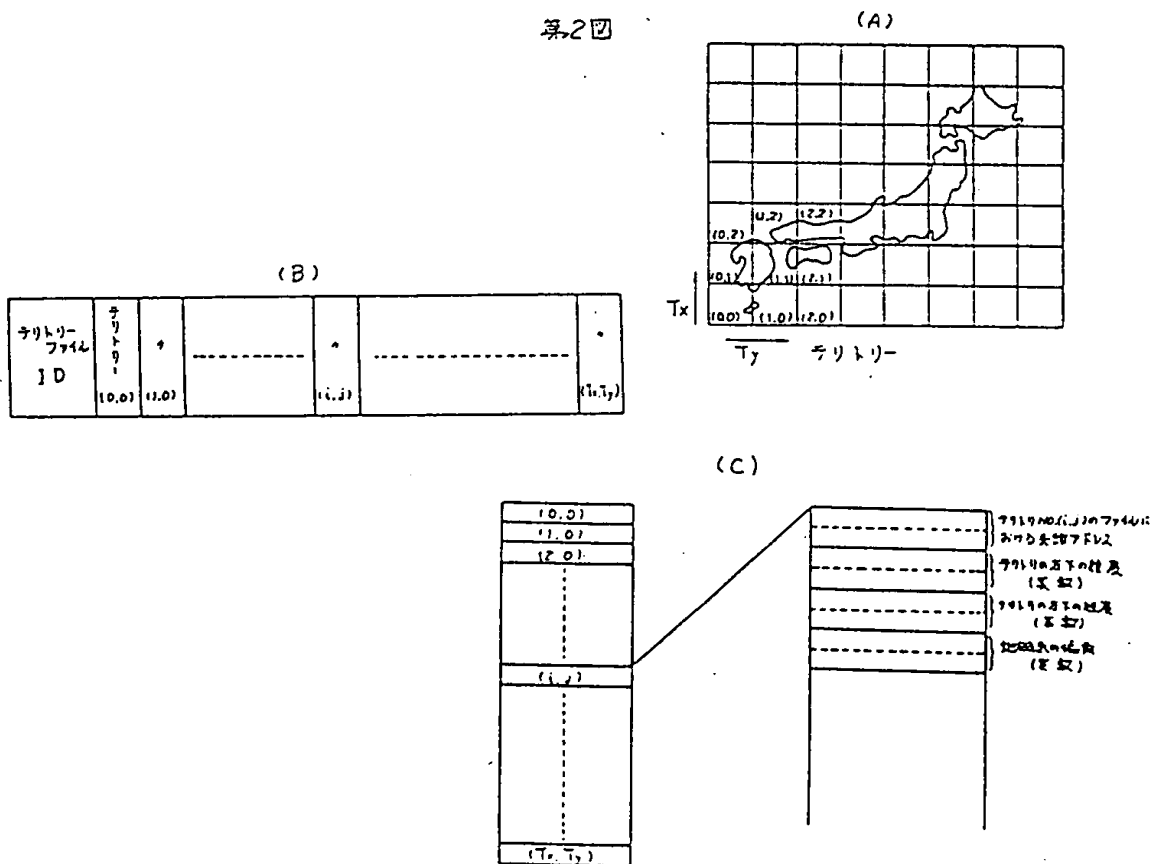
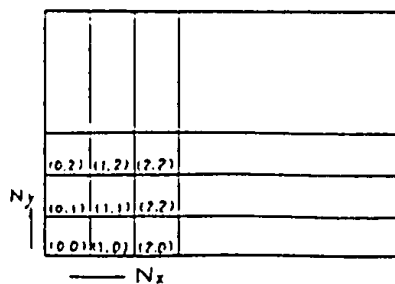
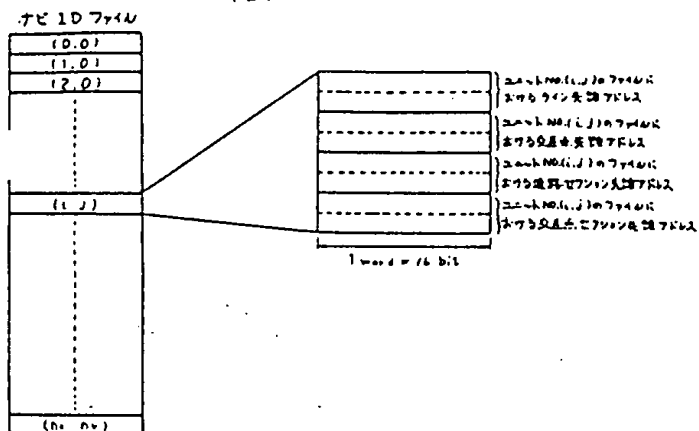
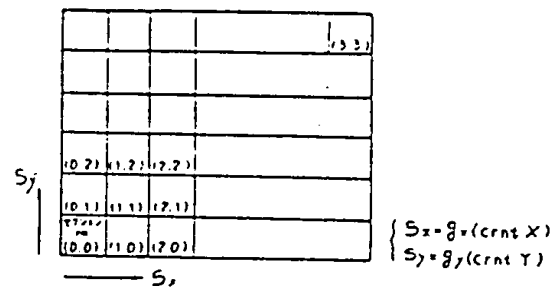




図 4 Z  
(A)

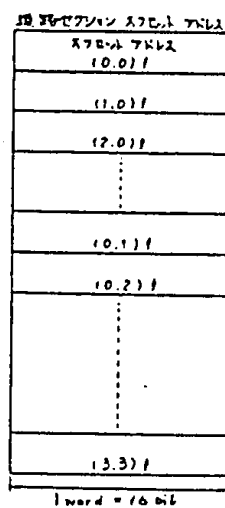
$$\begin{cases} N_x = \{x(\text{crnt } x)\} \\ N_y = \{y(\text{crnt } y)\} \end{cases}$$

(B)

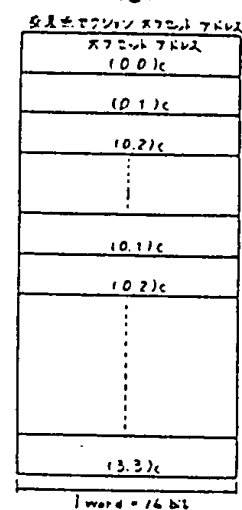
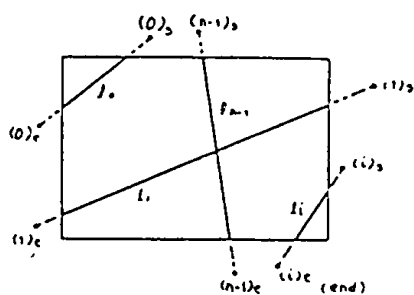
図 5 Z  
(A)

$$\begin{cases} S_x = \{x(\text{crnt } x)\} \\ S_y = \{y(\text{crnt } y)\} \end{cases}$$

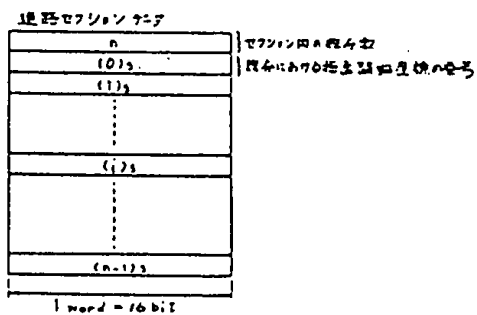
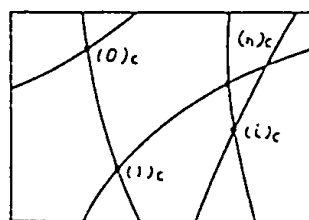
(B)



(C)

図 6 Z  
(A)

(B)

図 7 Z  
(A)

(B)

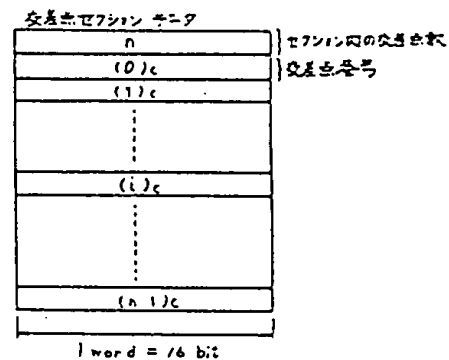
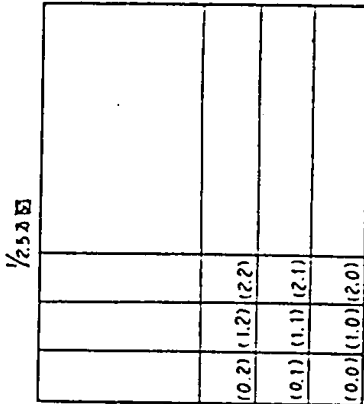


図8 (A)



$A_{0y}$

$$\begin{cases} A = K_{0x} (crnt\ x) \\ A = K_{0y} (crnt\ y) \end{cases}$$

(B)

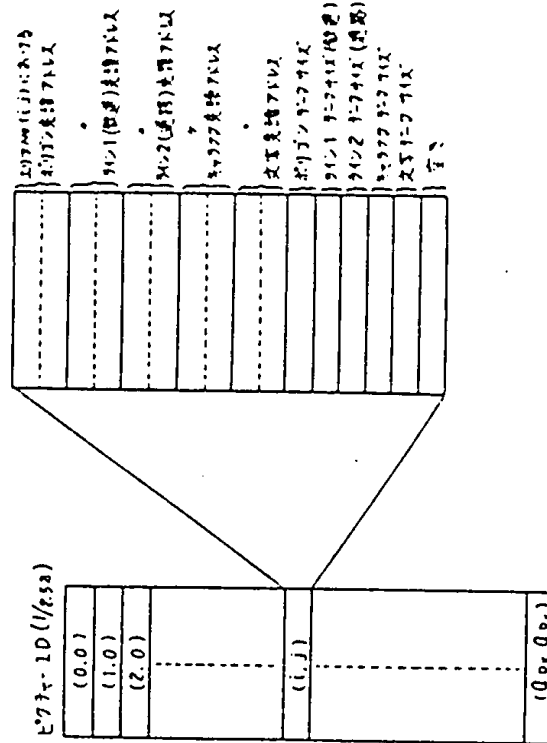
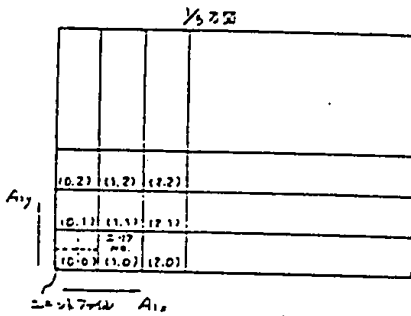
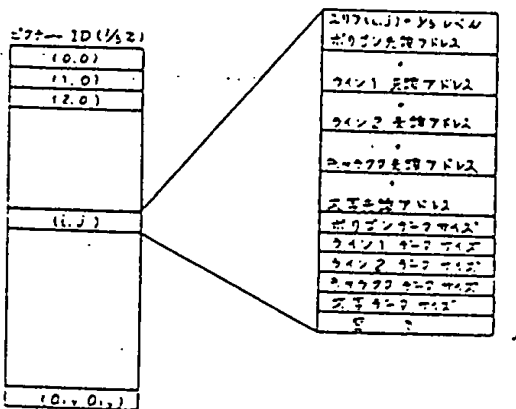


図9 (A)

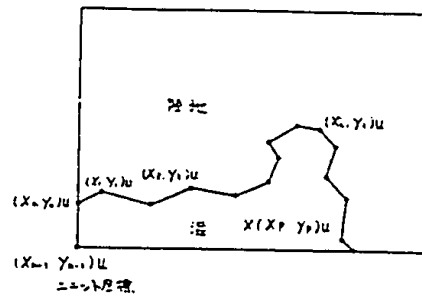


(B)

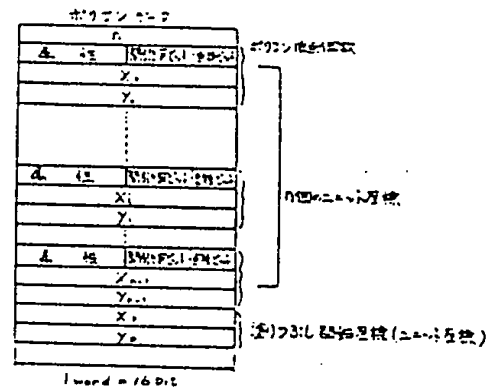


各ユニット命令

図11 (A)

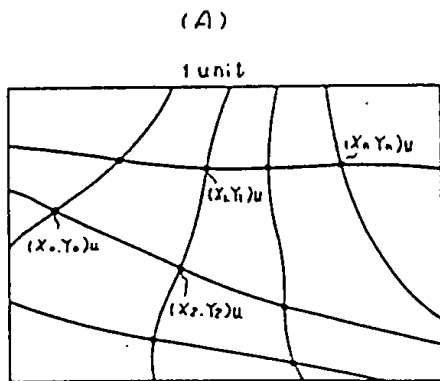


(B)

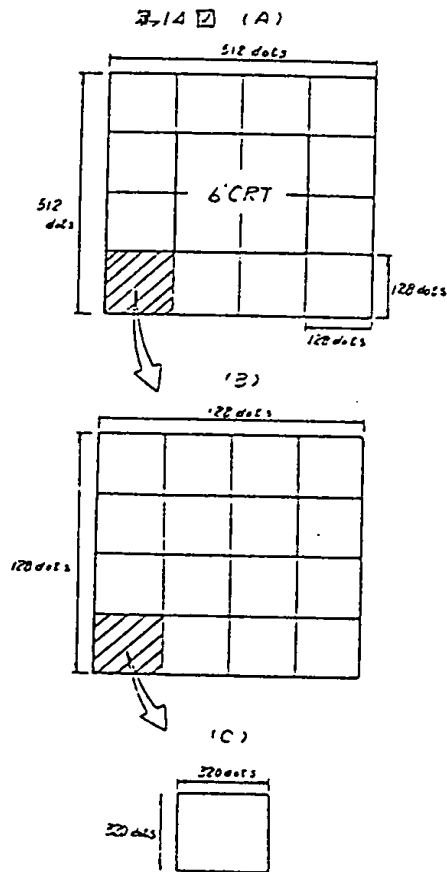
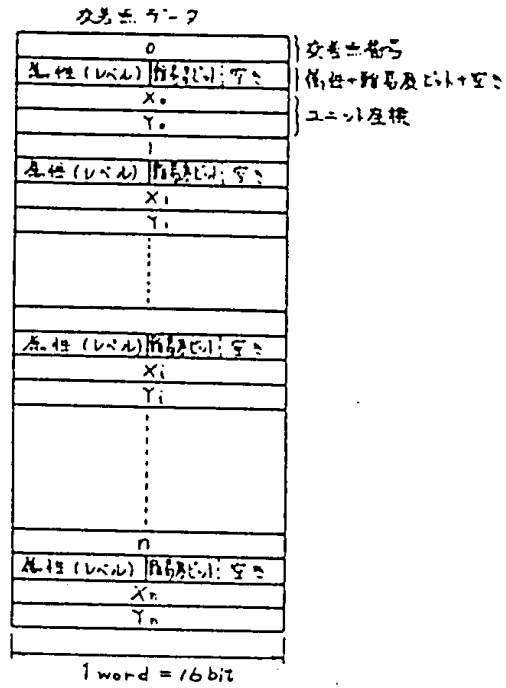




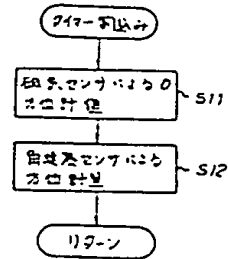
第13図



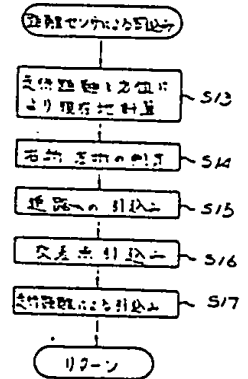
(B)



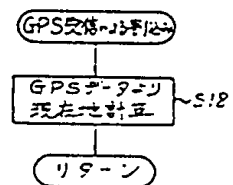
第16図



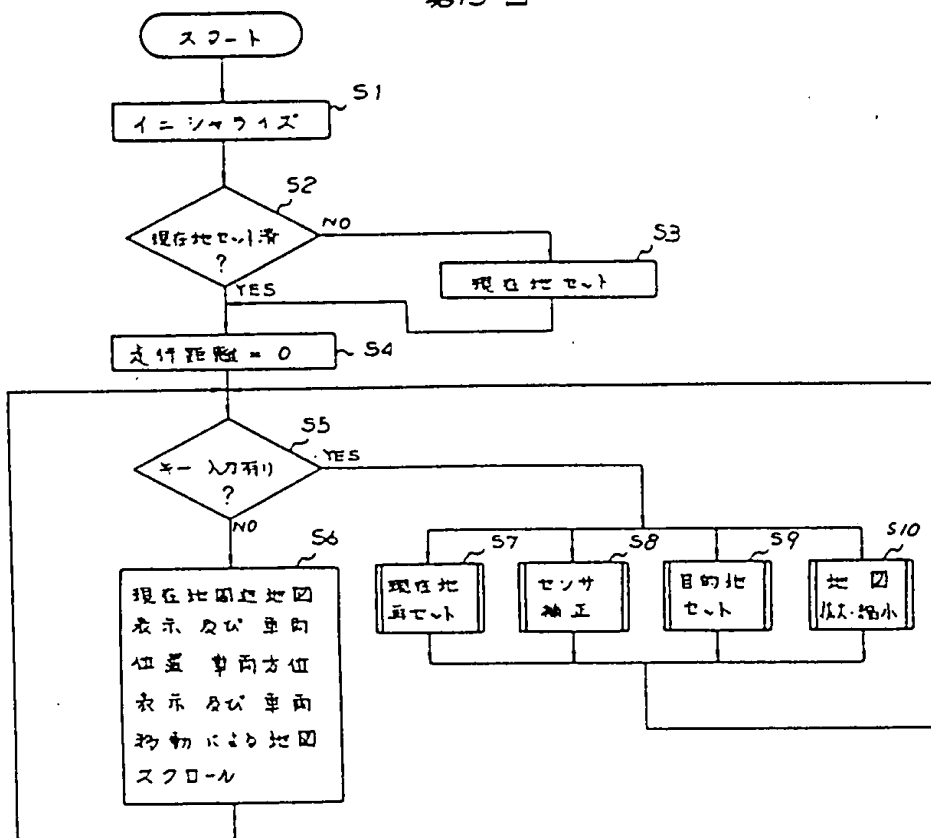
第17図



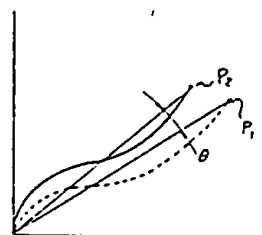
第18図



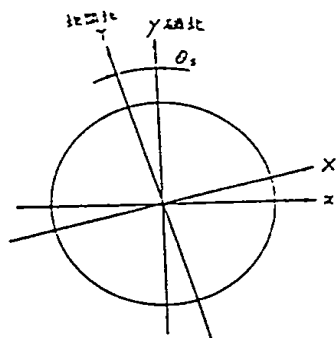
第15図



第19図

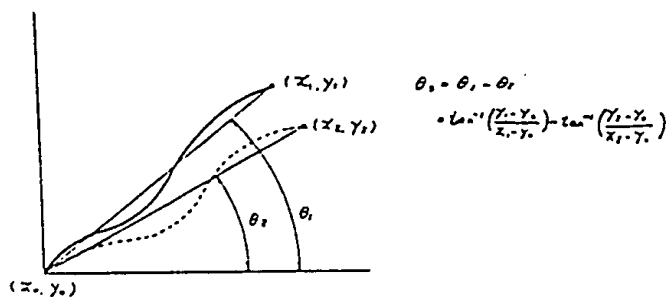


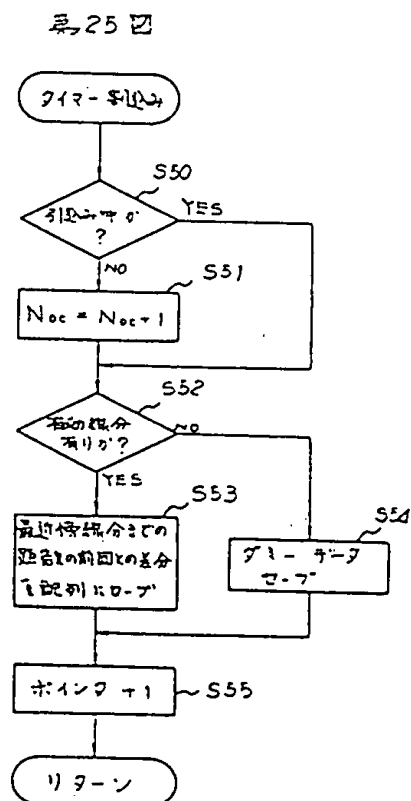
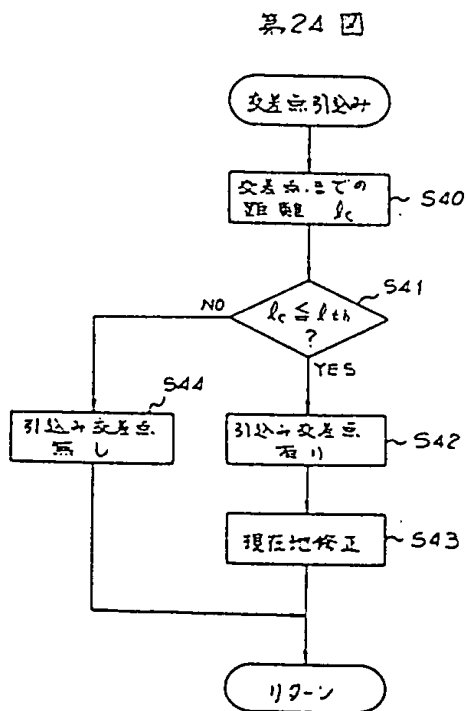
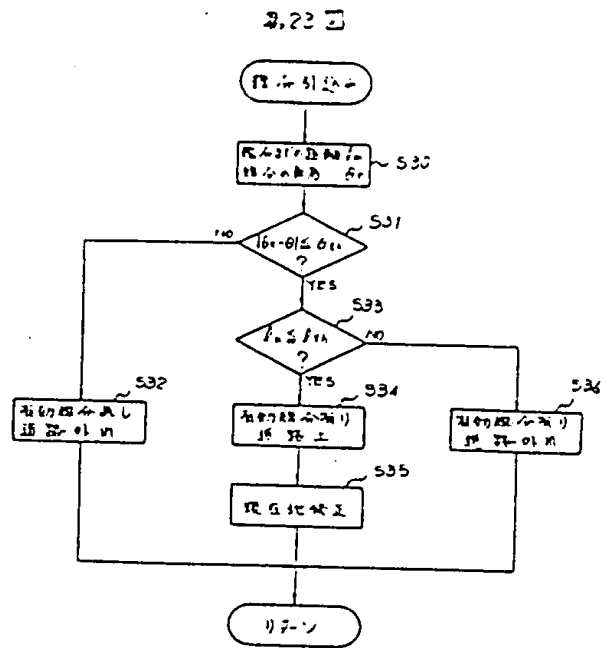
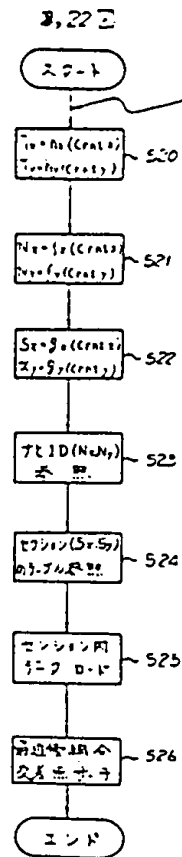
第20図



$$\begin{cases} X = x \cos \theta_s - y \sin \theta_s \\ Y = x \sin \theta_s + y \cos \theta_s \end{cases}$$

第21図





第26図

